Calhoun
Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1990-09

Measured noise performance of a data clock circuit derived from the local M-sequence in direct-sequence spread spectrum systems

Harshbarger, Stuart D.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/34897

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

AD-A238 335

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
ELECTE
JUL 16 1991
S B D

THESIS

MEASURED NOISE PERFORMANCE OF A DATA CLOCK CIRCUIT DERIVED FROM THE LOCAL M-SEQUENCE IN DIRECT-SEQUENCE SPREAD SPECTRUM SYSTEMS

by

Stuart D. Harshbarger

September, 1990

Thesis Advisor: Glen Myers

Approved for public release; distribution is unlimited.

91-04920

91 7 12 049

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) EC	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) MEASURED NOISE PERFORMANCE OF A DATA CLOCK CIRCUIT DERIVED FROM THE LOCAL M-SEQUENCE IN DIRECT-SEQUENCE SPREAD SPECTRUM SYSTEMS

12 PERSONAL AUTHOR(S)
HARSHBARGER, Stuart D.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1990 September	67

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Direct-sequence spread spectrum; communications; data clock recovery; M-sequence; delay-lock loop; spread spectrum; binary sequence generation

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

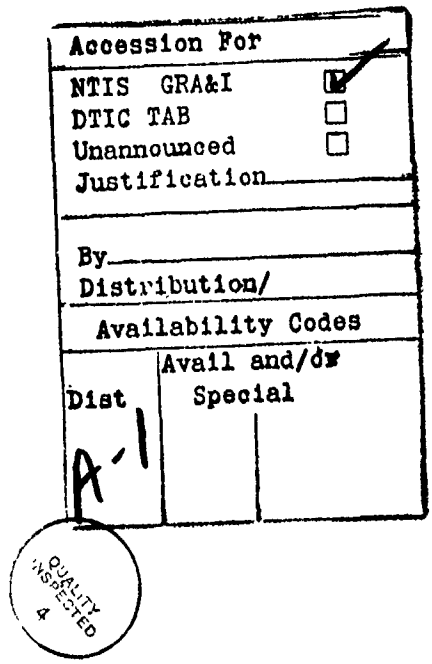
An improved method for deriving the timing information required for data recovery in the receiver of direct-sequence spread spectrum systems was implemented in hardware. This method uses a priori knowledge of the spreading sequence and its relation to the transmitted data to determine the precise beginning and end of data bits in the received signal. Testing of the hardware built for this research is concerned primarily with the performance of the circuit designed to provide the timing required to implement an integrate and dump circuit as a means of data recovery. A conclusion of this research effort is that a method exists for deriving the timing information required for data recovery from the locally generated m-sequence in the receiver. This method appears to be superior to alternative methods since the reference timing is derived from the locally generated m-sequence and is therefore isolated from the effects of additive noise in the channel. In addition to this improvement in noise performance, the new method is

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL MYERS, Glen	22b TELEPHONE (Include Area Code) 408-646-2325 22c OFFICE SYMBOL EC/Mv

DD Form 1473, JUN 86 *Previous editions are obsolete* SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

19. cont.
independent of transitions in the data stream which permits design flexibility for voltage representation of bits. Further, the timing may be derived from existing hardward that is a part of many direct-sequence spread spectrum communication systems.



Approved for public release; distribution is unlimited.

Measured Noise Performance of a Data Clock
Circuit Derived From the Local M-sequence in
Direct-Sequence Spread Spectrum Systems

by

Stuart D. Harshbarger
Electronics Engineer, Naval Air Test Center
B.S.E.E., West Virginia Institute of Technology, 1985

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1990

Author: [redacted]
Stuart D. Harshbarger

Approved by: [redacted]
Glen Myers, Thesis Advisor

[redacted]
Tri Ha, Second Reader

[redacted]
Michael Morgan, Chairman, Department of
Electrical and Computer Engineering

ABSTRACT

An improved method for deriving the timing information required for data recovery in the receiver of direct-sequence spread spectrum systems was implemented in hardware. This method uses *a priori* knowledge of the spreading sequence and its relation to the transmitted data to determine the precise beginning and end of data bits in the received signal. Testing of the hardware built for this research is concerned primarily with the performance of the circuit designed to provide the timing required to implement an integrate and dump circuit as a means of data recovery. A conclusion of this research effort is that a method exists for deriving the timing information required for data recovery from the locally generated m-sequence in the receiver. This method appears to be superior to alternative methods since the reference timing is derived from the locally generated m-sequence and is therefore isolated from the effects of additive noise in the channel. In addition to this improvement in noise performance, the new method is independent of transitions in the data stream which permits design flexibility for voltage representation of bits. Further, the timing may be derived from existing hardware that is a part of many direct-sequence spread spectrum communication systems.

TABLE OF CONTENTS

Page

I. INTRODUCTION 1

A. BACKGROUND 2

B. SUMMARY 3

C. CHAPTER CONTENTS 4

II. DIRECT-SEQUENCE SPREAD SPECTRUM 6

A. TRANSMITTER 6

B. PROPERTIES OF THE SPREADING SEQUENCE 8

C. RECEIVER 10

1. Code Tracking Loop 12

2. Data Recovery System 15

a. Data Recovery Methods 15

(1) Lowpass Filter Detection 15

(2) Integrate and Dump Detection 17

b. Data Clock Recovery Methods 19

III. STATEMENT OF THE PROBLEM 22

IV. EXPERIMENTAL CLOCK RECOVERY SYSTEM 24

A. TRANSMITTER 24

B. CHANNEL 25

C. RECEIVER 26

1. Code Tracking Loop 26

2. Data Recovery System 28

3. New Method for Deriving the Data Clock 29

4. Performance of the New Method 31

V. CONCLUSIONS AND RECOMMENDATIONS 39

A. CONCLUSIONS 39

B. RECOMMENDATIONS 39

LIST OF REFERENCES 40

APPENDIX A 41

APPENDIX B 49

APPENDIX C 56

INITIAL DISTRIBUTION LIST 58

ACKNOWLEDGMENTS

I would like to extend my appreciation to Professor Glen Myers for his technical guidance and enthusiasm throughout this effort. Also, I would like to acknowledge Mr. Chris Bartone of the Naval Air Test Center for his time during the many discussions of his previous research effort and for his recommendations regarding this effort.

I. INTRODUCTION

In a direct-sequence spread spectrum communication system, data modulates a high rate binary code in the transmitter to produce a transmitted bandwidth which is much larger than the minimum bandwidth required to transmit the data. Direct sequence spread spectrum systems are used because they exhibit an improvement in signal-to-noise ratio (SNR) which provides anti-jam/interference rejection capabilities. Moreover, because the signal is spread over a large bandwidth, the power density with respect to frequency is reduced which may provide for covert communications. In addition, proper signal design can provide for time/ranging information, or code division multiple access applications. [Ref. 1:p. 12]

Historically, direct-sequence spread spectrum systems were used primarily in military applications. Recently, however, these systems have found applications in a variety of commercial communication systems [Ref. 2].

Direct-sequence systems, like all digital communication systems, have a disadvantage in that data recovery in the receiver requires reference timing to determine where individual data bits begin and end in the demodulated waveform. This research concerns a new method of obtaining accurate timing for data recovery.

A. BACKGROUND

The motivation for this investigation began with thesis research conducted by Chris G. Bartone at the Naval Postgraduate School, Monterey, CA. As part of his research, a prototype direct-sequence spread spectrum system with a spreading factor of 255 was constructed, and its noise performance characteristics were measured. Since this system was designed only for experimental purposes, the timing information required to provide data recovery in the receiver was obtained via wire from the transmitter's data clock. [Ref. 1]

This previous effort resulted in a recommendation that such a system should be considered for implementation by the Department of Defense in AM radio systems to provide low data rate anti-jam/interference-rejection capability, primarily as an alternative to normal AM voice radio when rendered useless due to interference. As a result of this recommendation, the Naval Science Assistance Program initiated an investigation in conjunction with engineers at the Naval Air Test Center to determine the feasibility of employing such a system for use with existing AM equipment. This work identified that such a system would require integrate and dump data recovery to achieve the required performance. The timing information required for data recovery must be obtained from the received signal. Although there are several clock recovery methods

which may be implemented to recovery this timing information, these techniques are typically rather complex and often hamper receiver noise performance. In addition, these circuits often depend on the number of transitions in the data stream, so performance degrades during long sequences of ones or zeros.

The motivation for this research effort is to determine an improved method of deriving the timing information required for data recovery from the received signal in a practical direct-sequence spread spectrum system in which the maximum data rate is limited by constraints on the spreading clock rate and the required processing gain.

B. SUMMARY

As part of this effort, a direct-sequence spread spectrum system employing a dual-channel delay-lock loop was designed, constructed, and tested. This effort emphasizes the measured noise performance of the system. Testing concerns verification of the performance of a circuit designed to provide the timing information required to implement an integrate and dump circuit as a means of data recovery.

A spreading factor of 255 is implemented by using a maximal-length binary sequence generator. The system performance is measured at baseband only; however, the spreading clock rate is constrained to maintain compatibility with existing superheterodyne AM radio equipment in the spirit of the previous research efforts.

Bandlimited white Gaussian noise is added to the received signal, and the bit error rate is measured at several signal-to-noise ratios in order to assess the measured noise performance of the system. This performance is then compared to theoretical expectations and to the performance results of previous investigations.

A conclusion of this research effort is that a method exists for deriving the timing information required for data recovery directly from the locally generated m-sequence in the receiver. This method takes advantage of apriori knowledge of the spreading sequence to determine the precise beginning and end of the individual data bits. This method appears to be superior to alternative methods in terms of both performance and implementation simplicity.

C. CHAPTER CONTENTS

Chapter II is intended to provide readers who are not familiar with spread spectrum the background needed to understand the operation of a typical direct-sequence spread spectrum system. This chapter includes transmitter and receiver descriptions and a detailed explanation of the properties of the maximal-length sequence. The final section of Chapter II discusses several alternative methods of obtaining the timing information for data recovery and the trade-offs associated with these methods. Readers familiar with the topics that Chapter II addresses may procede directly

to Chapter III. Chapter III defines the problem of concern and indicates the nature of the research effort. Chapter IV presents details related to the design and operation of the system that was built including signal representations as required. Finally, Chapter V provides the conclusions and recommendations resulting from the research effort.

II. DIRECT-SEQUENCE SPREAD SPECTRUM

Spread-spectrum can be defined as a means of data transmission in which the transmitted signal occupies a bandwidth in excess of the minimum necessary to send the information. The increase in bandwidth (spread) in the transmitted spectrum is accomplished by means of a spreading sequence (code) that is modulated by the data. Recovery of the data in the receiver is accomplished by using the autocorrelation properties of the spreading sequence to synchronize a locally generated version of the spreading sequence with the received signal. This synchronized sequence is used for despreading the received signal and for subsequent data recovery.

A. TRANSMITTER

A typical direct-sequence spread-spectrum transmitter is shown in Figure 1. In such a transmitter, both the spreading clock and a much slower data clock are derived from a common oscillator (shown as Main Clock in Figure 1). Data then modulates the higher rate binary sequence from the spreading code generator to form a composite signal $V_{mc}(t)$. If the spreading code sequence is chosen such that one complete cycle is generated for each transmitted data bit, then the sequence itself is transmitted when the data signal $V_d(t)$ is a "one",

and the inverse of the sequence is transmitted when $V_d(t)$ is a "zero". In a practical system, the composite sequence $V_{mc}(t)$ then modulates a carrier for transmission over a channel.

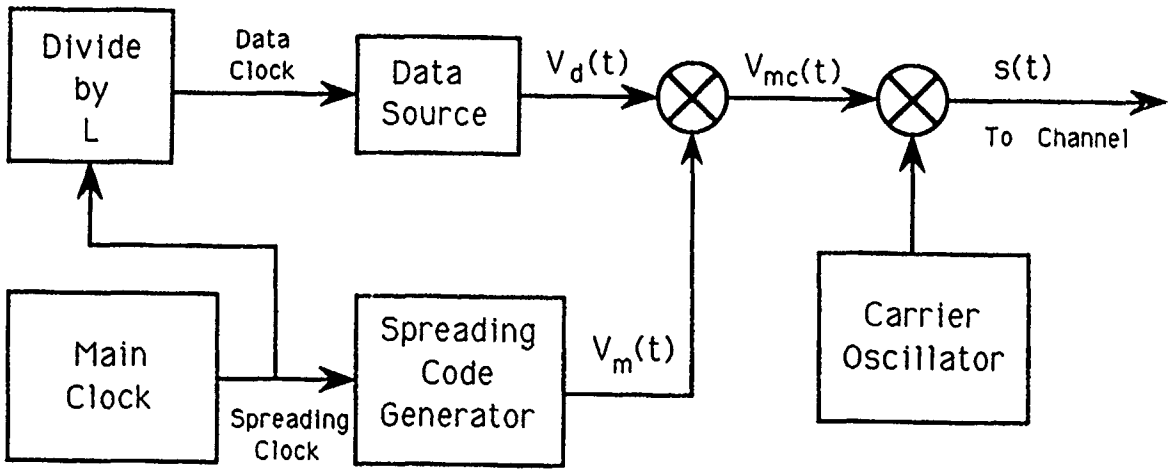


Figure 1. Typical Direct-Sequence Spread Spectrum Transmitter. After Ref. 1.

A popular choice for carrier modulation in a direct-sequence spread spectrum system is binary phase shift keying (BPSK). BPSK is often used to provide a signal of "constant" amplitude as it propagates through the channel. Further, it enables recovery of the modulation when the SNR is much less than unity [Ref. 1:p. 20].

B. PROPERTIES OF THE SPREADING SEQUENCE

The spreading sequence is usually chosen to be a maximal length binary sequence (m-sequence). An m-sequence is by definition the longest sequence that may be generated by an n-stage linear feedback shift register. The feedback taps which result in m-sequences for shift registers of various length are tabulated [Ref. 3: p. 67]. The length of an m-sequence is $L=2^n-1$ where n is the length of the sequence generator [Ref. 3:p. 28]. A realization of an m-sequence of length L=255 is shown in Figure 2.

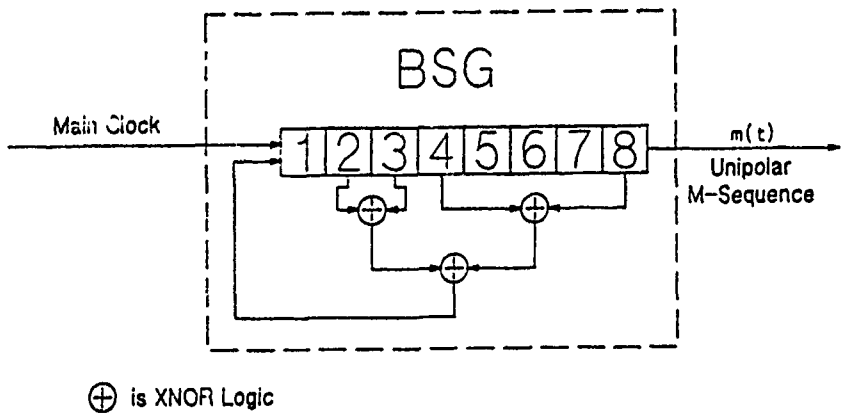


Figure 2. 8-bit Maximal Length Binary Sequence Generator (BSG) with Feedback Connections [8,4,3,2]. From Ref. 1.

The individual bits within a cycle of this spreading sequence are called "chips". A complete cycle of an m-sequence will contain 2^n-1 unique sequences of n bits each which will all lie on one cycle of length 2^n-1 [Ref. 4]. This cyclic property of the m-sequence will be discussed in more detail within the body of the report.

If the 8-stage BSG shown in Figure 2 is used as the spreading code generator in the transmitter shown in Figure 1, then a data "one" will be represented by a pseudo-random sequence of 255 chips. Likewise a data "zero" will be represented by the inverse of the 255 bit sequence. Since the data bits are transmitted at a rate equal to 1/L times the spreading sequence rate, or 1/255 times the data rate for the BSG shown, an entire cycle of the chip sequence will be generated for each data bit, and the beginning of each cycle will be synchronized with the beginning of the corresponding data bit.

An m-sequence also exhibits good autocorrelation properties. The autocorrelation $R_m(\tau)$ of an m-sequence with a constant amplitude of $\pm$ 1 volt can be shown to equal [Ref. 5]

$$R_m(\tau) = \begin{cases} 1-\frac{\tau}{T_c}\left(1+\frac{1}{L}\right) & \text{for } 0 \le \tau \le T_c \\ -\frac{1}{L} & \text{for } T_c < \tau < \frac{T_m}{2} \end{cases} \qquad (1)$$

where

T_c = duration of one chip

L = length of the sequence = number of chips in one period

T_m = period of the sequence = LT_c.

This autocorrelation function is an even function and is shown in Figure 3. If an m-sequence with the same feedback taps as that of the transmitter's BSG is generated locally in

the receiver subsystem, the characteristics of the autocorrelation function may be used to recover the transmitted data. This process is discussed in the following section.

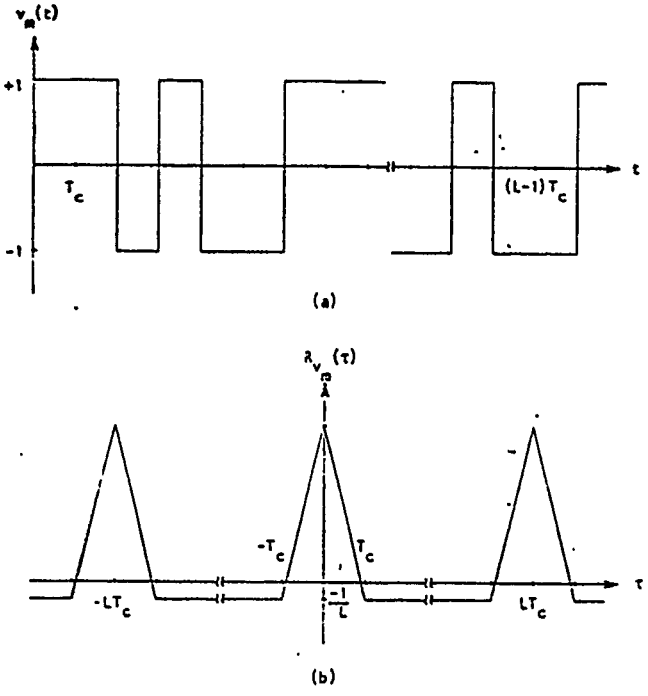


Figure 3. Illustration of a) M-Sequence of Length L and period T_c, and b) its Autocorrelation Function.

C. RECEIVER

A typical direct-sequence receiver is shown in Figure 4. The left hand portion of this figure illustrates a typical superheterodyne front end and demodulator to recover the baseband signal $V_r(t)$ plus additive interference or noise $n_r(t)$ from the channel. The baseband receiver subsystem consists of a code tracking loop and a data recovery system.

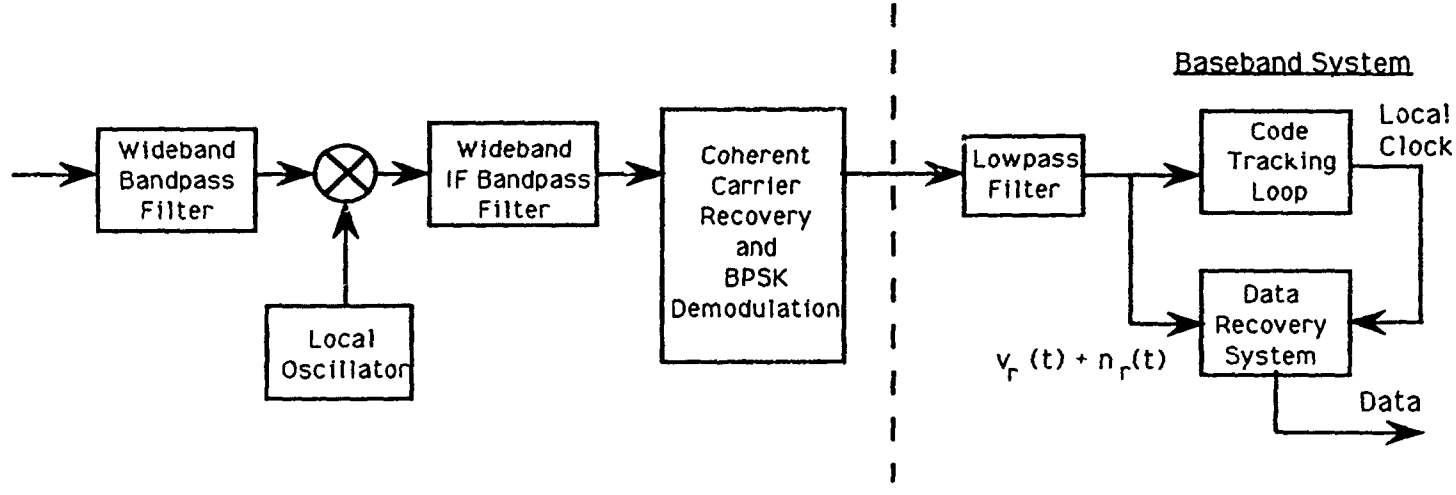


Figure 4. Typical Direct-Sequence Spread Spectrum Receiver. After Ref. 1.

1. Code Tracking Loop

The code tracking loop is responsible for synchronizing a locally generated replica of the transmitted m-sequence, with the baseband signal $V_r(t)+n_r(t)$. The dual-channel delay-lock loop (DCDLL) shown in Figure 5 is often implemented as the code tracking loop in direct-sequence receivers.

Since the received baseband signal is often buried in noise, the DCDLL uses the autocorrelation function (ACF) properties of the m-sequence to synchronize the locally generated sequence with the received composite sequence. Prior to system "LOCK", the frequency of the delay-lock loop's voltage controlled oscillator (VCO) is offset by biasing the voltage control input such that the output of the VCO is at a frequency higher than that of the transmitter's main clock. Three outputs are then taken from the m-sequence generator such that there is one clock delay between each tap. The center tap is passed on to the punctual correlator circuit in the data recovery system, while the first and third taps are applied to an "early correlator" and a "late correlator" respectively. The outputs of these correlators are passed through the ACF rectification circuits to remove inversions produced in the ACF waveform caused by modulation of the sequence by the data being transmitted. Since the delay-lock loop's VCO is biased to operate faster than the clock used to generate the incoming signal, the locally produced m-sequence

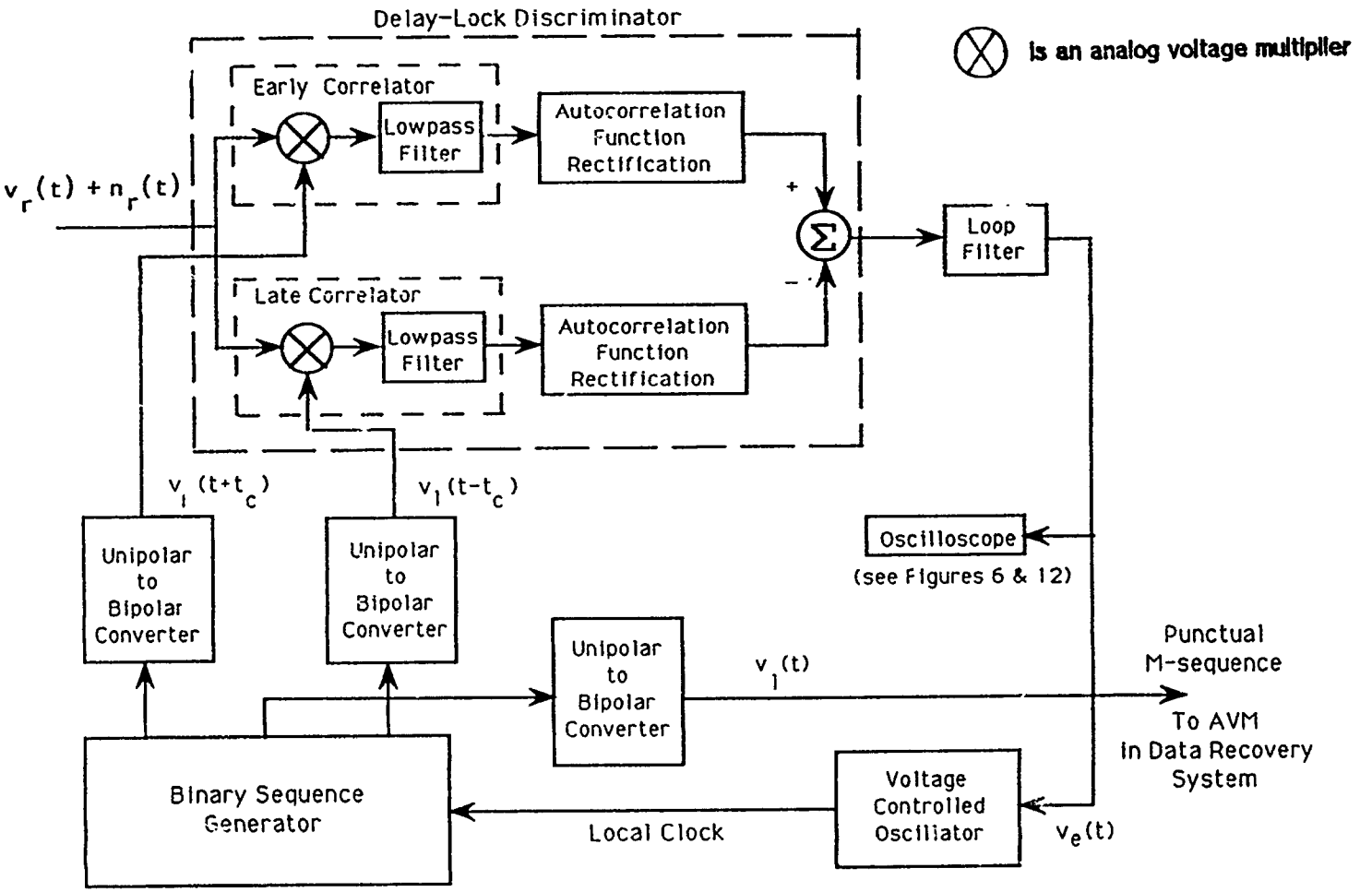


Figure 5. Typical Dual-Channel Delay-Lock Loop. After Ref. 1.

will effectively pass the incoming sequence, and the error voltage out of the delay-lock-loop will respond in the manner shown in Figure 6.

If the bias voltage to the receiver VCO is released near the center of the error waveform, and if the error voltage is appropriately filtered, the feedback loop will track the center portion of the error waveform to maintain alignment with the midpoint of the "early" and "late" taps; that is, the input sequence to the punctual correlator will be synchronized with the sequence containing the transmitted data that is embedded in $V_r(t)+n_r(t)$.

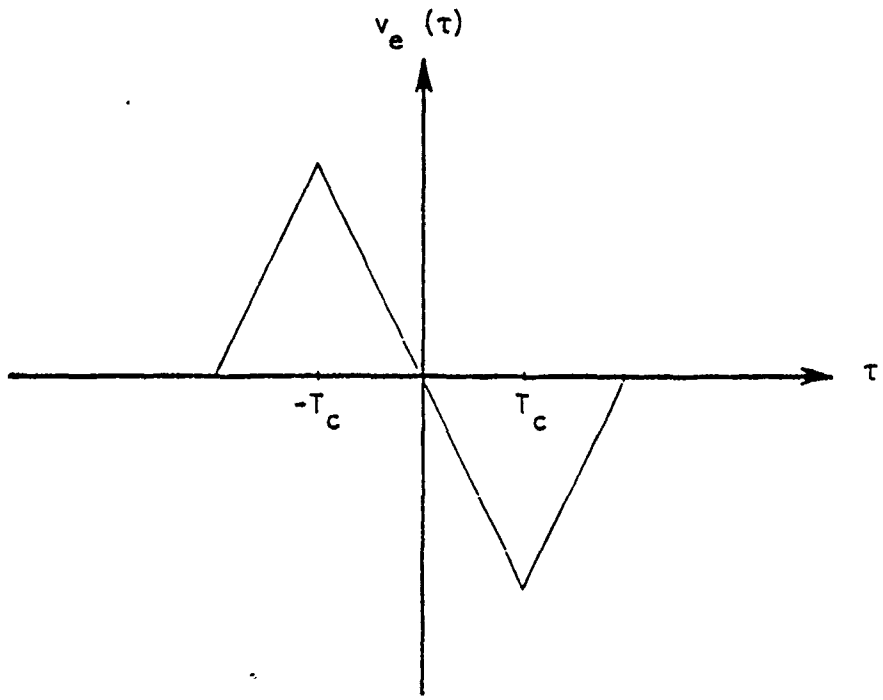


Figure 6. Typical Delay-Lock Loop Error Signal. From Ref. 1.

2. Data Recovery System

Once the signal from the delay-lock loop has been synchronized with the sequence in $V_r(t)+n_r(t)$, the two signals may be multiplied by an analog voltage multiplier (AVM). At this point, the transmitted data signal, corrupted by additive noise from the channel, has been separated from the received composite signal. The function of the data recovery system is to recover the individual data bits from the noisy signal at the output of this "punctual AVM". See Figure 7.

a. Data Recovery Methods

In practice there are two methods which may be used to detect the individual data bits in the signal at the AVM output. These are lowpass filter or matched filter detection. The equivalent of a matched filter for a polar rectangular pulse shape is an integrator [Ref. 6:p. 528].

(1) Lowpass Filter Detection. For lowpass filter detection the signal at the output of the punctual AVM is first applied to a lowpass filter, then a sample and hold device and a threshold detector are used at the filter output to recover the individual data bits. The equivalent bandwidth B of the lowpass filter (LPF) is chosen such that $B > 1/T$, where T = the width of an individual data bit. This value of cutoff is chosen so that the signal waveform is preserved and the noise is reduced. If the polar NRZ signal (See Figure 8) of interest is passed through a unity gain LPF with a cutoff

of B Hz, and then applied to a threshold detector with an optimum setting of $V_T=0$ (i.e. a zero crossing detector), the probability P_e that a bit is detected in error is found to be [Ref. 6:p.530],

$$P_e = Q\left(\sqrt{\frac{A^2}{N_o B}}\right) \qquad (2)$$

where N_o = power spectral density (PSD) of additive white Gaussian noise at the input of the detector.

For lowpass filter detection, a data clock is required for control of the sample and hold timing. Although this technique for detecting the individual bits is easy to implement, an improvement in system performance can be obtained by integrate and dump detection.

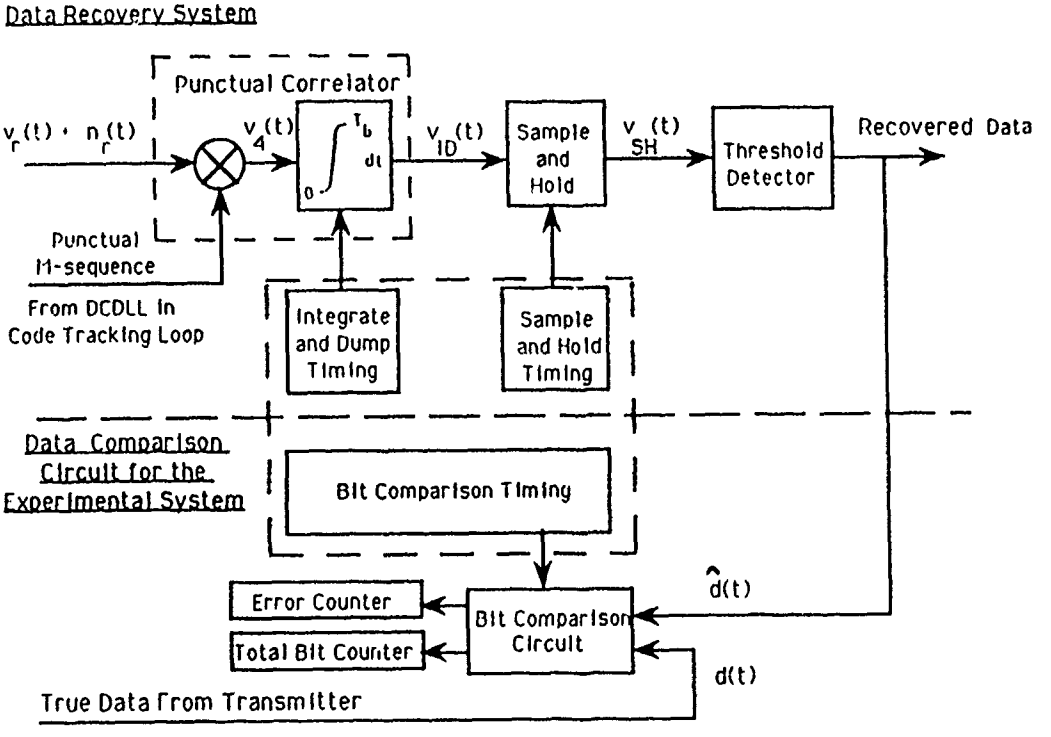


Figure 7. Data Recovery System Block Diagram. From Ref. 1.

(2) Integrate and Dump Detection. Since the signal at the output of the punctual correlator is strongly influenced by additive noise introduced in the channel, the SNR should be maximized at the sampling instant to obtain the best estimate of each individual data bit. For a digital waveform, only the information content represented by the value of a particular bit is of interest, and there is no requirement to preserve the shape of the waveform. An integrate and dump circuit can be used to maximize the SNR at the sampling point. The output of the integrate and dump circuit is then applied to a sample and hold circuit and a threshold detector. For a polar NRZ signal, integrate and dump detection results in a P_e of,

$$P_e = Q\left(\sqrt{\frac{2A^2}{N_o B}}\right) \qquad (3)$$

which provides approximately 3dB of P_e performance over that of the lowpass filter technique [Ref. 6:p. 531]. The integrate and dump detector, like the lowpass filter technique, can be easily implemented except that it requires reference timing to synchronize the sample and dump pulses with the individual data bits in the received signal. Assuming that this reference information is available, the signal is integrated over each bit interval. The constant value of the bit results in either a positive going ramp (data "one") or a negative going ramp (data "zero") at the

integrator output. Since the noise has a zero mean characteristic, it tends toward zero over each bit period. The remaining ramp can be sampled near the end of the bit interval to determine the value of the bit. Once the bit has been sampled, the integrator must be dumped so that is ready for the following bit interval. A data recovery system for a direct-sequence spread spectrum system which employs integrate and dump detection is shown in Figure 7. It is clear that a reliable method of obtaining the required reference timing is necessary to provide optimization of system noise performance. The following section addresses several methods of acquiring this reference signal.

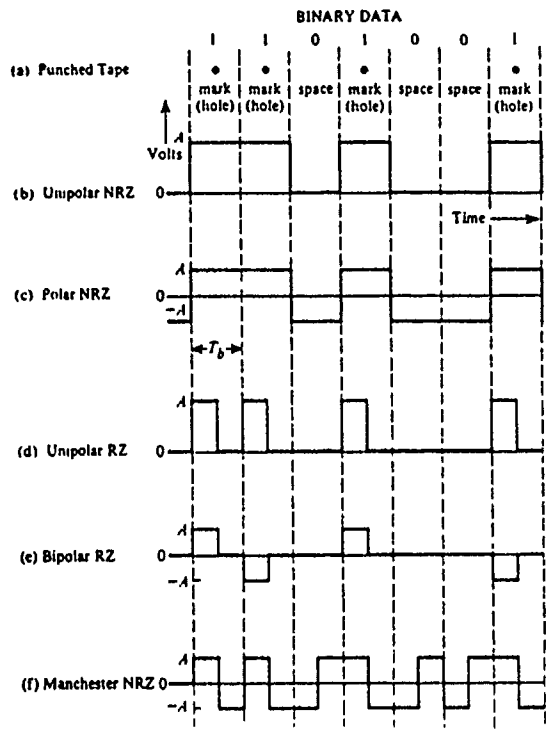


Figure 8. Binary Signaling Formats. From Ref. 6.

b. Data Clock Recovery Methods

Current techniques that may be employed to obtain the required timing information in the receiver include the following:

(a) The reference signal may be transmitted on a separate carrier, or the reference may be encoded in the data by means of time-of-day information provided to both the transmitter and receiver by an accurate time reference.

(b) Transitions in the data signal may be used to extract the reference signal from the data stream itself.

This research effort is concerned primarily with systems which obtain this reference information from the received data signal. In such a system, transitions in the data signal occur as a result of clocking by the system oscillator. If an alternating stream of "ones" and "zeros" are transmitted, data transitions will occur at a rate directly related to that of the system clock. The data clock can then be recovered in the receiver by lowpass filtering the signal at the output of the punctual AVM, and then applying the signal to a threshold detector. This signal can then be applied to a narrowband phased-lock-loop (PLL) circuit that serves to eliminate false transitions introduced by the additive channel noise. This ensures that the transitions occur at a rate restricted to a given tolerance.

Unfortunately the transmitted data occurs as a random sequence of zeros and ones. The result is that several data intervals may occur without a transition in the data value. If too many data intervals occur without a transition, the PLL circuitry cannot maintain lock, and the PLL output frequency may begin to drift. The synchronization error introduced by this drift directly affects system noise performance, since errors in the integrate and dump timing directly reduce the SNR at the sampling point.

One practical solution that ensures at least one transition during each bit interval is to use a manchester NRZ line code (See Figure 8). Use of this line code ensures that the PLL receives transitions at a rate sufficient to maintain "lock". The disadvantage of using a manchester line code is that twice the number of level transitions are required to transmit the same amount of data. This means that either the data must be transmitted at 1/2 the polar NRZ transmission rate, or that the system bandwidth must be doubled [Ref. 7]. For a system in which the maximum data rate is limited by constraints on the maximum clock rate and the required processing gain, the use of manchester encoding would further reduce the maximum data rate by a factor of two. This may be unacceptable.

Another method of data clock recovery for polar NRZ signalling involves a technique known as an early-late bit synchronizer is shown in Figure 9 [Ref. 6:p. 158]. This

technique operates in a manner similar to that of the DLL of the receiver's code tracking loop. This method involves a considerable amount of complexity, and is also dependent upon a sufficient number of data transitions for reliable operation.

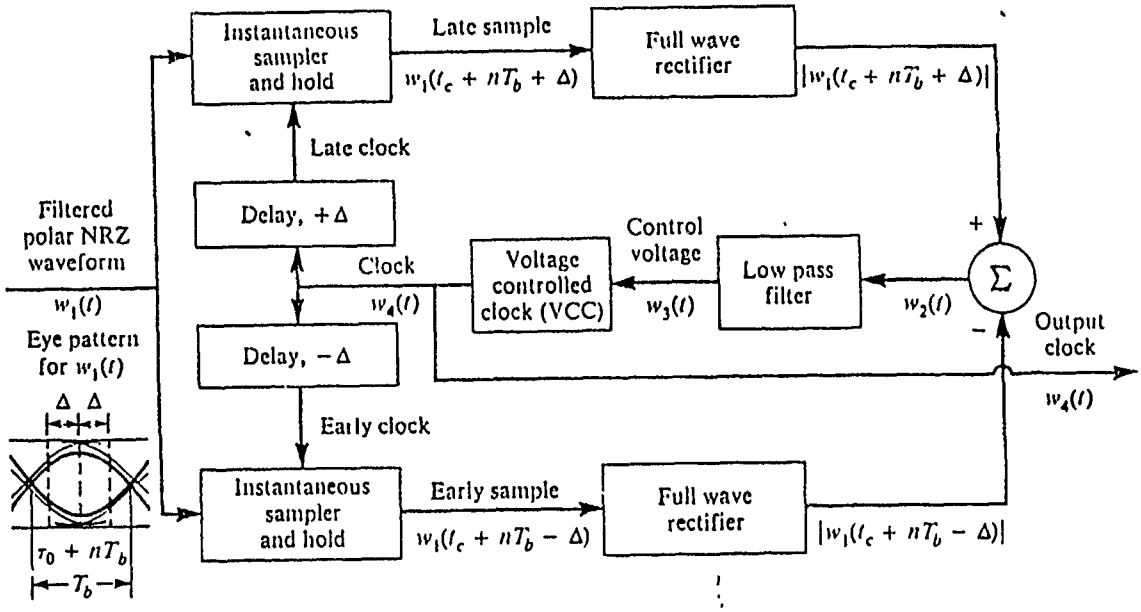


Figure 9. Early-Late Bit Synchronizer. From Ref. 6.

A method which may be employed to ensure reasonable data transitions without an increase in system bandwidth requirements is a technique known as interleaving. In this technique, the transmitted data is scrambled so that the data stream contains a sufficient number of data transitions. The data is then unscrambled to its original order prior to the receiver output.

III. STATEMENT OF THE PROBLEM

There is a need for an improved method of obtaining the reference timing required for data recovery in direct-sequence spread spectrum receivers. Each of the methods which are currently used to obtain this reference timing obtain a data clock from the received data signal. Since this received data signal may be corrupted by additive noise from the channel, the performance of each of these methods degrades as the SNR decreases. The performance of such methods also depends on the number of transitions in the data stream, and the circuitry required to ensure that clock transitions occur only at the appropriate times is often rather complex.

This research effort seeks to determine an improved method for deriving the data clock in the receiver of direct-sequence spread-spectrum systems. As part of the research an experimental direct-sequence transmitter and receiver were designed and constructed. Several experimental circuits for deriving a data clock were attempted. These failed to provide any improvement over existing methods. Finally, a new method for deriving the required timing information was conceived. This new method exploits apriori knowledge of the transmitted composite signal to extract the required timing information from the locally generated m-sequence, once it has been

aligned with the incoming sequence by the code tracking loop. This new method was incorporated into the experimental system, and the measured noise performance of the system was determined. It was found that the proposed method provides an optimum means of deriving the required timing information, because the information is derived from the locally generated sequence in the receiver, and is therefore isolated from additive channel noise and transitions in the data stream. The design of the experimental system, along with the theoretical justification for the proposed method, are discussed next.

IV. EXPERIMENTAL CLOCK RECOVERY SYSTEM

During the course of this investigation, a method was conceived which takes advantage of the cyclic properties of the m-sequence and the relation between data and the local m-sequence in the receiver to identify the precise beginning and end of each data bit. To verify the performance of this method, a direct-sequence transmitter and a receiver utilizing this method were constructed and the measured noise performance of the system was determined.

A. TRANSMITTER

The direct-sequence transmitter shown in Figure 1 was implemented with a main clock of 19.125 kHz, and a spreading factor of 255. The spreading code was chosen to be a m-sequence with feedback taps of [8,4,3,2]. This m-sequence is generated using a PAL22V10 programmable logic device to implement the 8-bit linear feedback shift register as shown in Figure 2. A second PAL22V10 is used to divide the main clock by 255, resulting in a data rate of 75 Hz. Random data is generated by using an additional m-sequence generator with feedback taps of [8,5,3,1]. The unipolar data and spreading sequences are converted to bipolar signals by using standard RS-232 drivers, and then the following multiplication is

performed with an AD-534 analog voltage multiplier to form the baseband composite signal $V_{mc}(t)$. Figure 10 shows the relation between the data signal $V_d(t)$ and the composite sequence $V_{mc}(t)$.

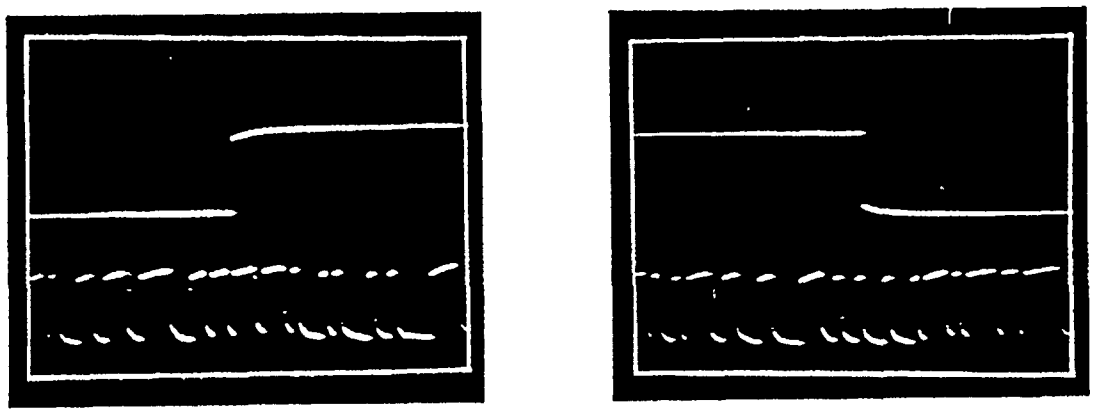


Figure 10. Relation between Data and the Spreading Sequence.

Once $V_{mc}(t)$ is formed, it is passed through a 20 kHz lowpass filter to make the waveform at the receiver input more representative of the received baseband signal in a practical system.

B. CHANNEL

To determine the measured noise performance of the experimental system, white Gaussian noise from an Elgenco model 603A WGN generator is added to the transmitted signal just prior to the receiver input. Figure 11 shows two cases for the received signal $v_r(t)+n_r(t)$.

Figure 11. Receiver Baseband Signal $v_r(t) + n_r(t)$. No Additive Noise (top) and SNR = -13 dB (bottom)

C. RECEIVER

The baseband receiver sub-system includes the code tracking loop and the data recovery system shown in the right-hand portion of Figure 4 on page 11.

1. Code Tracking Loop

The code tracking loop used in this investigation is the dual-channel delay-lock loop as shown in Figure 5 on page 13. Used in the receiver is a commercially available voltage controlled oscillator (VCO) with a rest frequency of 19.125 kHz. The output of this VCO is used to clock the BSG which generates three time-shifted replicas of the m-sequence used as the spreading sequence in the transmitter as described in Section C of Chapter II. The early and late taps are multiplied against the received signal by two AD534 AVMs. The

output of these two AVMs are then filtered by individual LTC1062 switched capacitor filter circuits designed with cut-off frequencies of 75 Hz each. The resulting signals are then rectified with two LM-741 operational amplifiers connected in a super-diode configuration, and then summed to form the error signal $v_e(t)$. A typical photograph of $v_e(t)$ for the open loop case is shown in Figure 12.

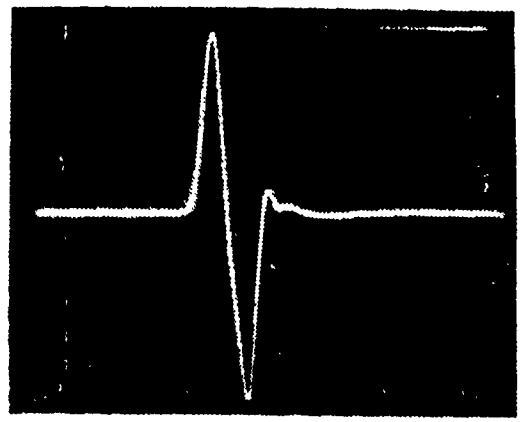


Figure 12. Typical DCDLL Error Signal $v_e(t)$. From Ref. 1.

In order to decrease synchronization time, the loop filter is designed such that a bias of 5 volts is applied to the VCO input when the system is not synchronized. This bias provides a difference between the main clock rate in the transmitter and the local clock rate. This allows the locally generated sequences to effectively pass the received waveform at a faster rate, thereby reducing acquisition time. Once the positive peak due to the early correlation function in Figure 12 is detected, the bias voltage is removed, and the VCO is returned to its rest frequency of 19.125 kHz, offset only by

the error voltage $v_e(t)$. In the closed loop configuration, $v_e(t)$ tracks the center portion of the waveform shown in Figure 12, and therefore ensures continued alignment of the locally generated "punctual" sequence with the received composite waveform.

2. Data Recovery System

A block diagram of a data recovery system employing a correlation detector is shown in Figure 6. The locally generated "punctual" sequence from the BSG in the DCDLL is multiplied against the received signal $v_r(t)+n_r(t)$ by a AD534 AVM to obtain the signal $v_d(t)$ which contains the demodulated data plus additive noise from the channel. To recover the individual data bits, it is desired to integrate over each individual bit period, and to sample the resulting signal near the end of each period. As discussed in Chapter II, this results in maximum SNR at the sampling instant and provides for optimum system noise performance. Figure 13 shows the signal $v_d(t)$ and the desired signal $v_o(t)$ which is obtained if the timing information required to dump the integrator is available for each data bit. The following section describes how the proposed method can be used to obtain the required timing information.

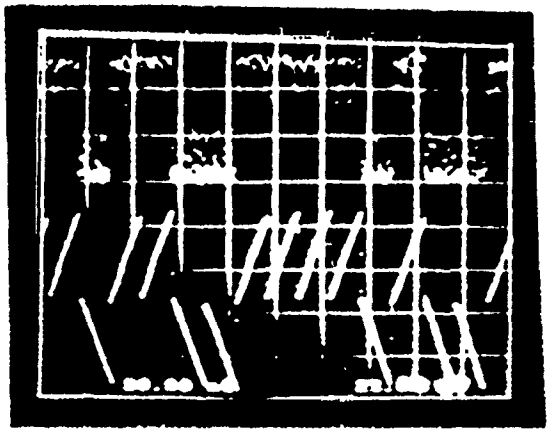


Figure 13. Punctual AVM Output $v_4(t)$ (top) and required Integrator Output $v_{id}(t)$ (bottom) for no additive noise.

3. New Method For Deriving the Data Clock

Both Figure 1 and Figure 10 show the relation between the data and the composite sequence in the transmitter. In these figures it can be seen that an entire cycle of the spreading m-sequence (or its inverse) occurs for each data bit. In the previous section, it was shown that the DCDLL ensures synchronization between the received composite signal and the locally generated punctual sequence, once the loop achieves lock. Since this locally generated m-sequence is aligned with the received signal, the position within any individual data bit may be found by determining the corresponding position within the cycle of the locally generated sequence.

The cyclic properties of an m-sequence can be used to determine the exact location within a cycle of the m-sequence, and consequently within the individual data bits. An m-sequence contains 2^n-1 unique pseudo-randomly distributed patterns that occur over the sequence's 2^n-1 cycle. These unique patterns may be visualized by sliding an n chip window along the entire cycle of 2^n-1 chips. Each of these patterns is unique in that it occurs only once during each cycle and the corresponding bit interval. Further, the cyclic properties of the m-sequence assure that this same n-chip pattern will occur 2^n-1 chip intervals later at the same position within the following data bit. [Ref. 4]

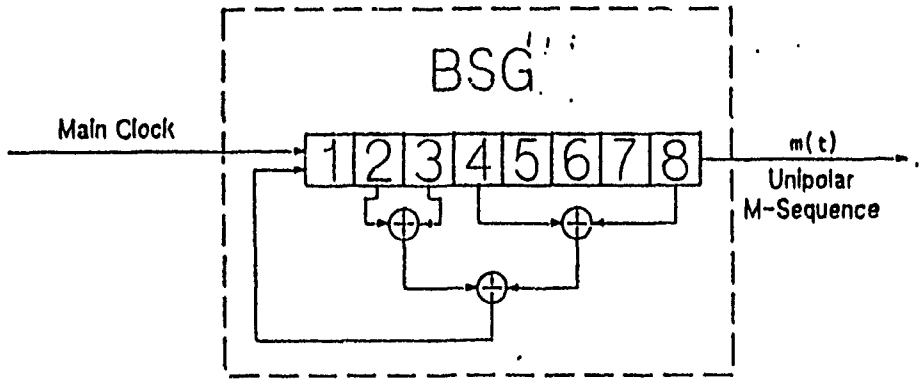
For the system constructed in this experiment, a data bit is represented by an entire 255 chip cycle of the spreading sequence, or the inverse of the 255 chip cycle. Once the locally generated m-sequence has been aligned with the received sequence by the DCDLL, the position within each data bit may be determined by monitoring the local sequence through an 8 chip window until the pattern corresponding to the desired location within the data bit occurs. Figure 14 shows the relation between the parallel outputs of the BSG from Figure 2 and the unipolar m-sequence. Although the 8 chip pattern appearing at the parallel outputs is not identical to the 8 chip patterns that appear in the output chip sequence, it can be shown that the 8 chip parallel patterns must each represent a unique state to ensure the cyclic properties of

the output sequence. It follows that any location within a bit interval can be flagged by simply detecting the 8 chip pattern at the parallel output that occurs at the appropriate instant. The 8 chip patterns for any given m-sequence can be easily generated with a computer algorithm once the system's feedback taps have been determined. Appropriate timing pulses may then be generated by simply adding combinational logic at the parallel output to produce the desired signals upon detection of the appropriate states.

4. Performance of the New Method

A MATLAB program was written to generate the output listing for the 8 stage BSG used in the experimental system (See Appendix A). It was found that the combinational logic required to obtain the sample and dump pulses required for reference timing in the data recovery system could be incorporated into the existing PAL22V10 used as the BSG in the DCDLL. The ABEL input file used to generate the PAL program is listed in Appendix B.

Figure 15 shows the sample and dump pulse timing with respect to a transition in the data stream and the integrator output. From this figure the precision of the sample and dump pulses relative to the end of a bit period is apparent.



⊕ is XNOR Logic

1	2	3	4	5	6	7	8	State #
1	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	2
0	1	1	0	0	0	0	0	3
1	0	1	1	0	0	0	0	4
1	1	0	1	1	0	0	0	5
1	1	1	0	1	1	0	0	6
1	1	1	1	0	1	1	0	7
0	1	1	1	1	0	1	1	8
1	0	1	1	1	1	0	1	9
0	1	0	1	1	1	1	0	10
1	0	1	0	1	1	1	1	11
1	1	0	1	0	1	1	1	12
0	1	1	0	1	0	1	1	13
.	.	.	.	.	.	.	.	.
0	0	1	1	1	1	0	1	249
0	0	0	1	1	1	1	0	250
0	0	0	0	1	1	1	1	251
0	0	0	0	0	1	1	1	252
0	0	0	0	0	0	1	1	253
0	0	0	0	0	0	0	1	254
0	0	0	0	0	0	0	0	255

Figure 14. Derivation of the Data Clock from the Local M-sequence.

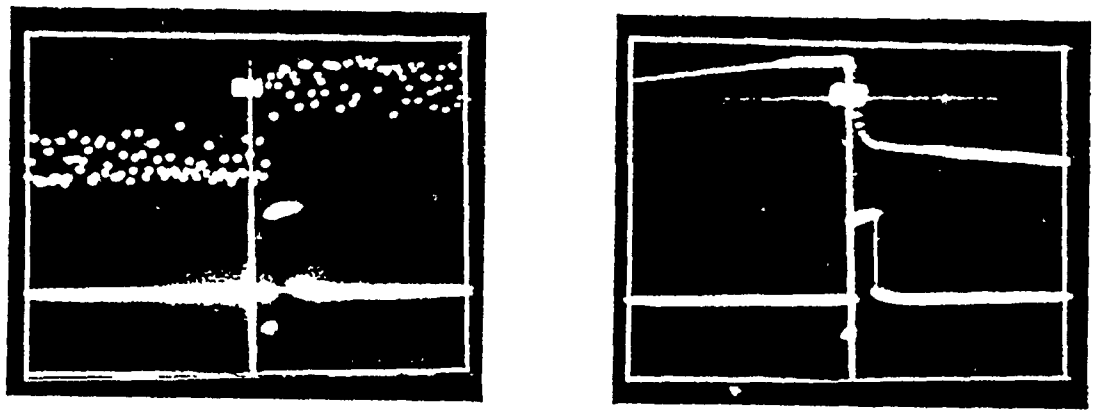
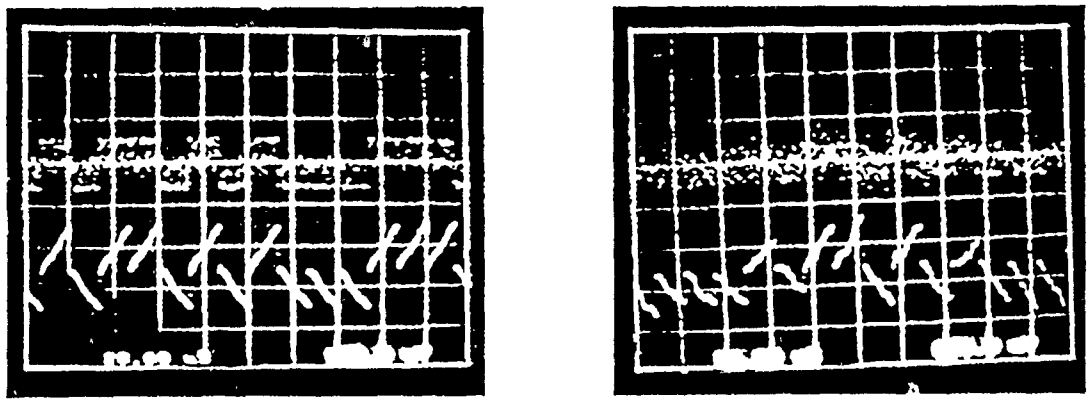


Figure 15. Timing of Sample and Dump Pulses with respect to Punctual AVM and Integrator Outputs.

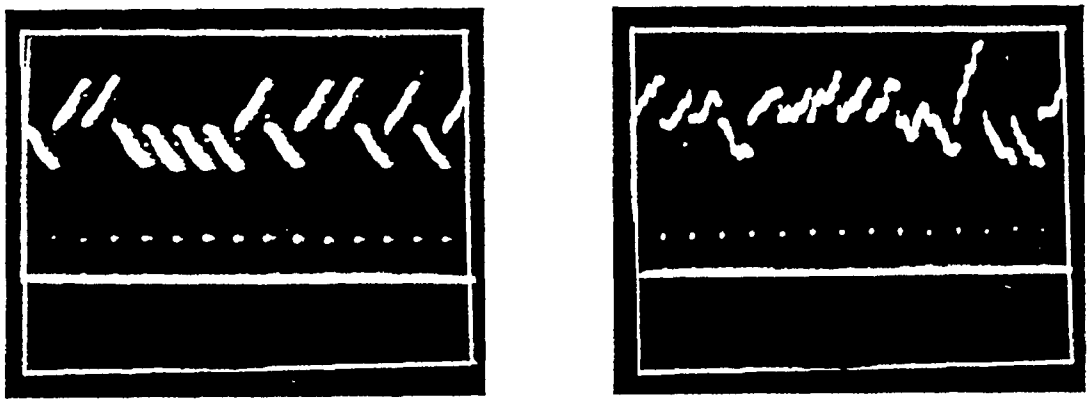
The waveform at the integrator output in shown in Figure 16 for two cases of input SNR. It can be seen that although the signal at the punctual AVM output may be completely buried in the additive noise from the channel, the integrator can still provide a signal from which the value of the individual data bits may be determined.



(a) (b)

Figure 16. Punctual AVM and Integrator Outputs, a) No additive noise, and b) SNR = -13 dB.

The relation of the dump timing to the integrator output is shown in Figure 17 for two cases of input SNR. Note that



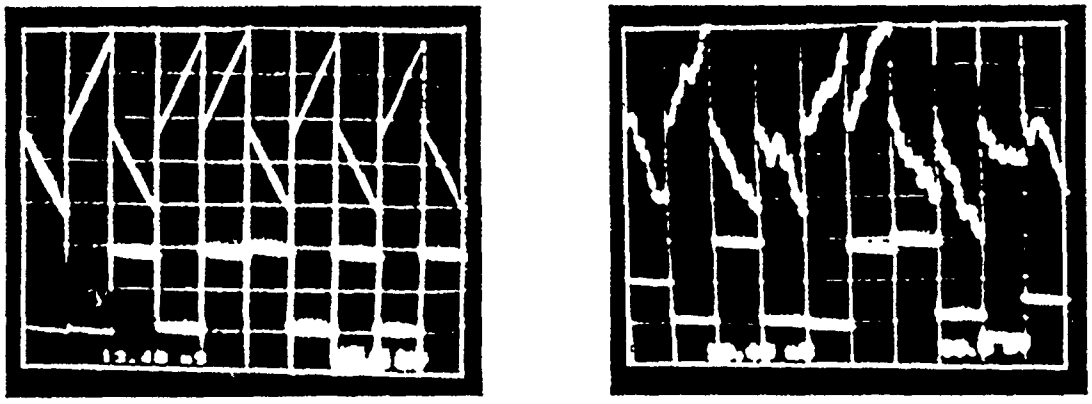
(a) (b)

Figure 17. Dump Timing with respect to Integrator Output, a) No additive noise, and b) SNR = -18 dB.

there is no jitter in the dump pulse timing even though the input SNR = -18 dB for the second case. Figure 18 shows the timing relation between the sample and hold output and the integrator output for two cases of input SNR. From this figure it can be seen that the integrator output is sampled precisely at the end of the bit interval for both cases of input SNR.

The stability in both the sample and the integrate and dump pulse timing results from the derivation of the timing from the locally generated m-sequence. The derivation of the reference timing from this local sequence isolates the reference signals from the effects of additive channel noise in the data stream. Since the reference timing is derived

from this locally generated sequence, an accurate data clock is available as long as the DCDLL maintains lock.



(a) (b)

Figure 18. Sample and Hold Output with respect to Integrator Output, a) No additive noise, and b) SNR = -18 dB.

The output of the sample and hold circuit is then applied to the zero crossing detector to finalize the data recovery process. To determine the system performance the recovered data is then compared to the true data from the transmitter as shown in Figure 19.

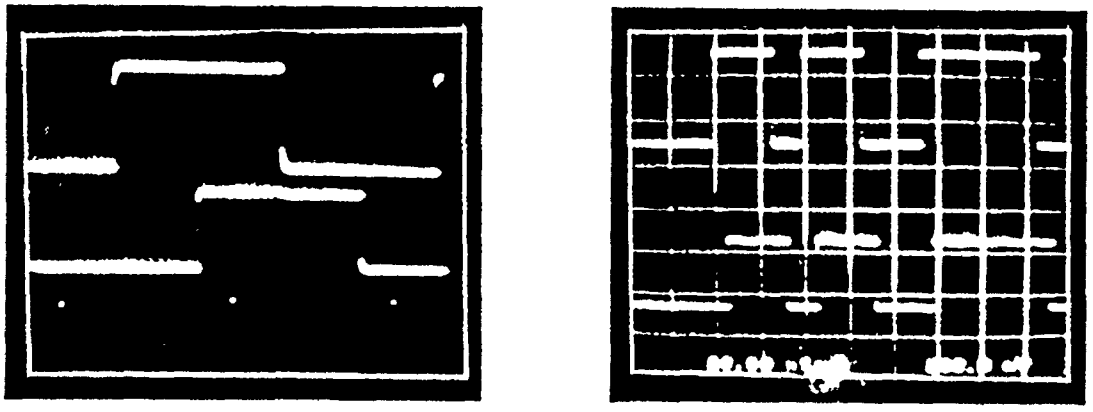


Figure 19. Data Comparision Timing and Recovered Data Stream.

The bit comparison pulse is generated in the same manner as the sample and dump pulses at an instant chosen near the midpoint of the overlapping data bits. The number of errors is then compared to the total number of bits transmitted (as shown in Figure 8) for several values of input SNR. The measured probability of error performance for the experimental system is shown in Figure 20. This measured noise performance is then shifted by the theoretical processing gain of 23.5 dB for the system and compared to the theoretical performance of a coherent BPSK system (See Figure 21).

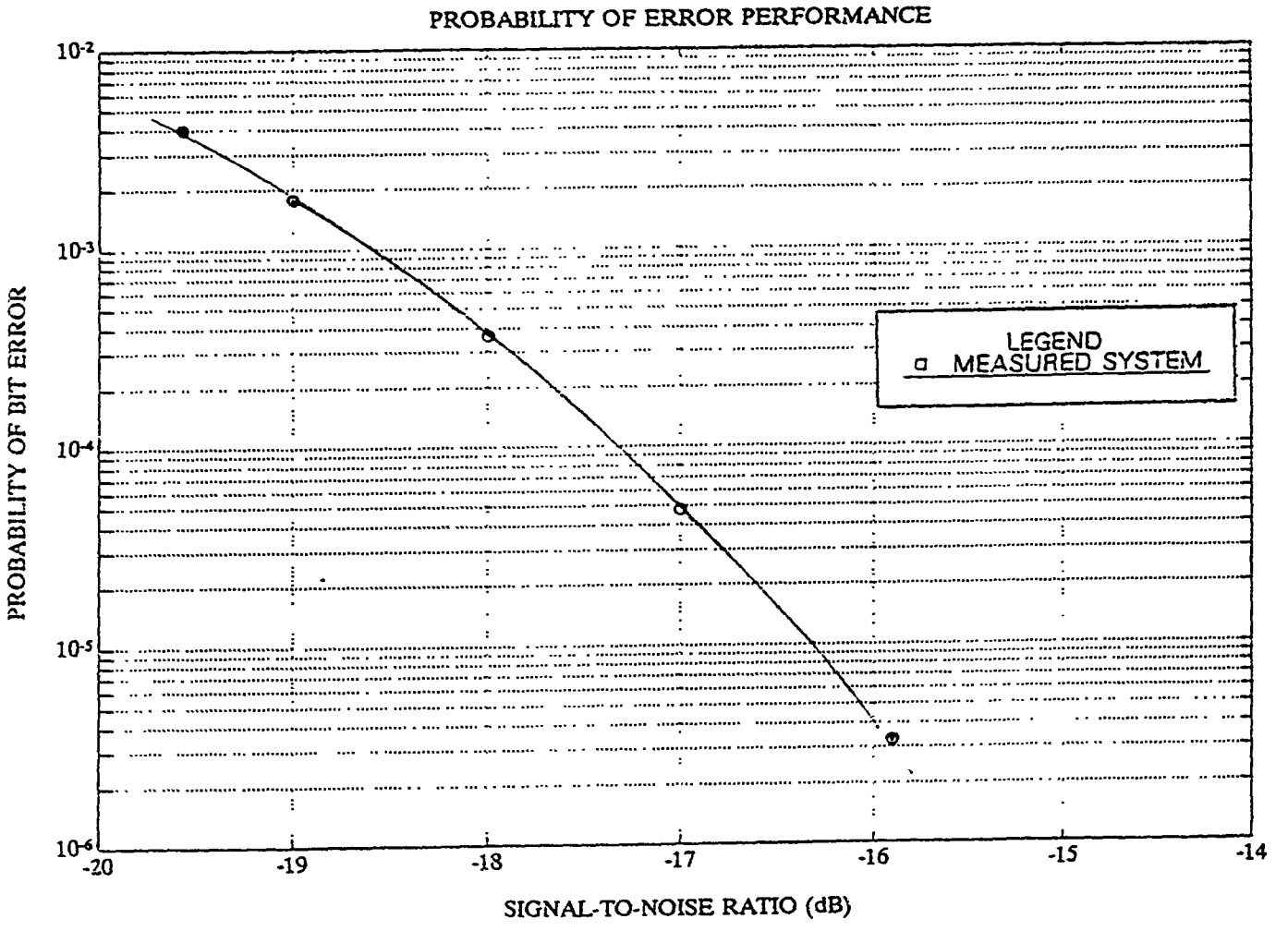


Figure 20. Probability of Bit Error Performance for the Experimental System.

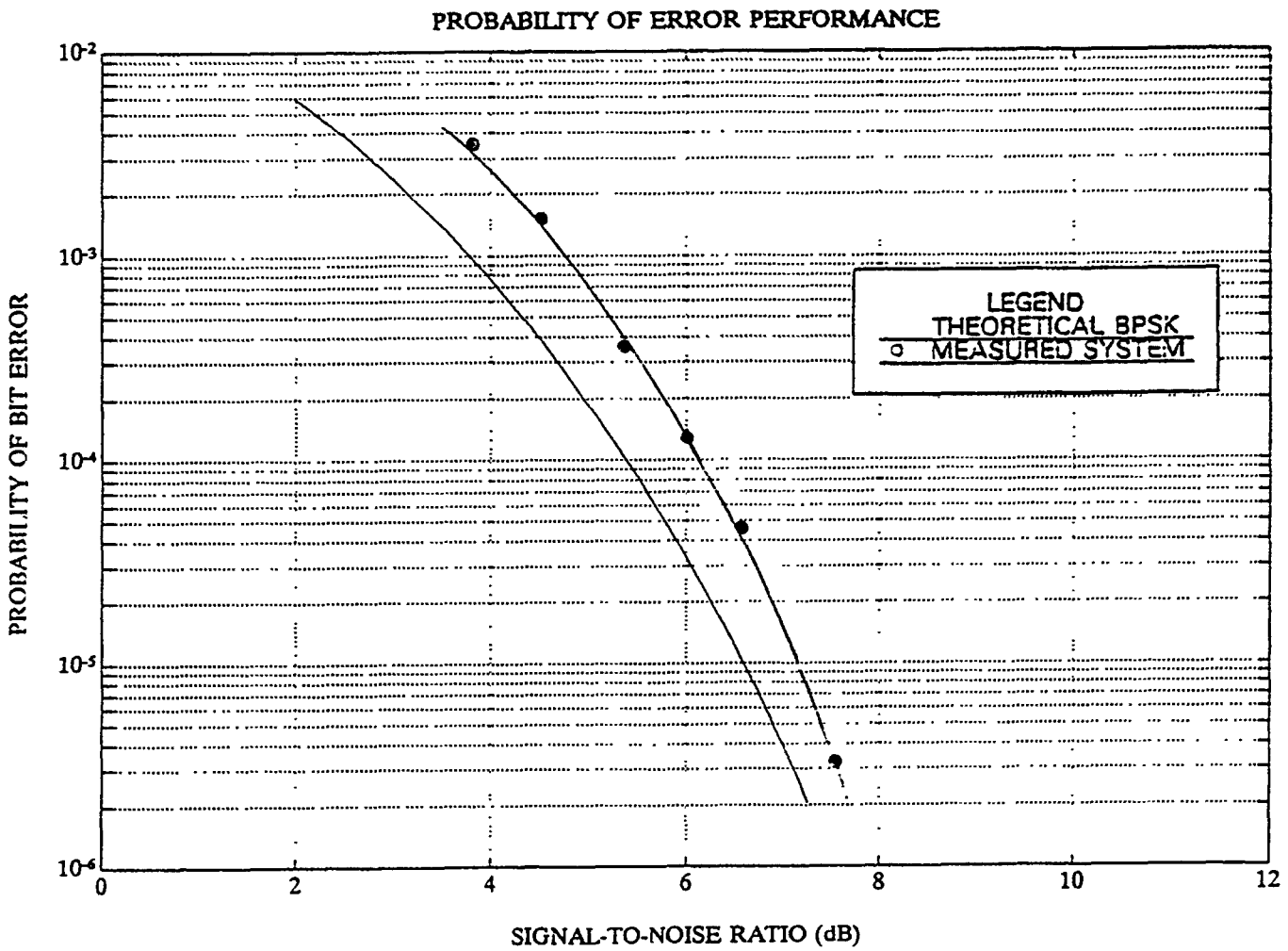


Figure 21. Probability of Bit Error Performance for the Experimental System shifted by Theoretical Processing Gain of 23.5 dB and Compared to Theoretical BPSK Bit Error Performance Curve.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

A new method for deriving the data clock from the locally generated m-sequence in the receiver of direct sequence spread spectrum systems was successfully demonstrated.

This method is superior to current methods for several reasons:

- The data clock is derived from a locally generated "clean" sequence, and is isolated from the effects of additive channel noise.
- Availability of accurate timing depends only on the receiver's ability to maintain "lock". While locked, the DCDLL continually ensures optimum alignment of the local sequence with the received composite sequence containing the data.
- The clock derivation is independent of transitions in the data stream.
- The circuit required to derive the required timing information can be implemented by modifying the firmware in the programmable logic device used to construct the local BSG.

B. RECOMMENDATIONS

It is recommended that the new method for deriving the data clock introduced in this research effort be considered for application to direct-sequence spread spectrum systems for all military and commercial applications which incorporate an m-sequence as the spreading code.

LIST OF REFERENCES

1. Bartone, Chris G., "Measured Noise Performance of a Direct-Sequence Spread Spectrum Communication System with a Comparison of Single Vice Dual-Channel Delay-Lock Loops", Master's Thesis, Naval Postgraduate School, Monterey, California, 1987.

2. Lee, Edward A. and Messerschmitt, David G., *Digital Communication*, p. 445, Kluwer Academic Publishers, Boston, Massachusetts, 1990.

3. Dixon, Robert C., *Spread Spectrum Systems*, 2d ed., John Wiley & Sons, Inc., New York, 1984.

4. Beker, H. J., *Secure Speech Communications*, pp. 112-113, Academic Press, San Diego, California, 1985.

5. Ziemer, Roger E. and Peterson, Roger L., *Digital Communications and Spread Spectrum Systems*, p. 387, Macmillan Publishing Company, 1985.

6. Couch, L. W., *Digital and Analog Communications Systems*, 2d ed, Macmillan Publishing Company, 1987.

7. Lathi, B.P., *Modern Digital and Analog Communication Systems*, CBS Publishing Company, New York, 1983.

APPENDIX A

The following is a listing for the MATLAB program which is used to obtain the states at the BSG outputs that correspond to the appropriate location within the spreading sequence. The program as listed produces a state table of length 255 for the M-sequence corresponding to feedback taps of [8,4,3,2].

Program listing:

```
QD=[0 0 0 0 0 0 0 0];
for j = 1:5;
    j, test=0; QD, TOTAL=0;
    for k = 0:254;

        % XOR BSG OUTPUTS (2&3) & (4&8) RESPECTIVELY

        if (QD(2)+QD(3)) == 1; A=0; else A=1; end;
        if (QD(4)+QD(8)) == 1; B=0; else B=1; end;
        if A+B == 1; C=0; else C=1; end;

        % GENERATE NEXT STATE

        Q=QD; QD(1)=C; QD(2)=Q(1); QD(3)=Q(2); QD(4)=Q(3);
        QD(5)=Q(4); QD(6)=Q(5); QD(7)=Q(6); QD(8)=Q(7);

        if (QD(1)+QD(2)+QD(3)+QD(4)+QD(5)+QD(6)+QD(7)+QD(8))
        ==0;..
        test=test+1; end;

        % STORE EACH STATE AS TABLE ENTRY

        r=k+1;
        M(r,1)=QD(1);    M(r,2)=QD(2);    M(r,3)=QD(3);
        M(r,4)=QD(4);
        M(r,5)=QD(5);    M(r,6)=QD(6);    M(r,7)=QD(7);
        M(r,8)=QD(8);
        M(r,9)=k+1;
```

```
% TEST TO ENSURE EACH POSSIBLE 8-BIT PATTERN WAS GENERATED

% Test for total = summation of integers from zero to 254.
        if QD(1) == 1; TOTAL=TOTAL + 1; end;
        if QD(2) == 1; TOTAL=TOTAL + 2; end;
        if QD(3) == 1; TOTAL=TOTAL + 4; end;
        if QD(4) == 1; TOTAL=TOTAL + 8; end;
        if QD(5) == 1; TOTAL=TOTAL + 16; end;
        if QD(6) == 1; TOTAL=TOTAL + 32; end;
        if QD(7) == 1; TOTAL=TOTAL + 64; end;
        if QD(8) == 1; TOTAL=TOTAL + 128; end;
    end;
    M, test, TOTAL,
    check=0;
    for k = 1:254; check =check + k; end;
    check, pause;
end;
```

The following table is a list of the 255 states generated from the MATLAB program listed above. The vertical columns represent the time shifted m-sequences which appear at each of the BSG's parallel outputs. Three of these outputs are chosen for the early, late, and punctual m-sequences in the dual channel delay-lock-loop. The state of the parallel outputs which corresponds to a particular instant within a data bit may be determined by searching the column that is associated with the output from which the punctual m-sequence is taken until the sequence associated with the beginning of the data bit is found. Any location within the data bit (or the spreading sequence) can then be determined by adding the appropriate offset to the reference state number.

Q(1)	Q(2)	Q(3)	Q(4)	Q(5)	Q(6)	Q(7)	Q(8)	Ref. #
QD= 0	0	0	0	0	0	0	0	-0-
M =								
1	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	2
0	1	1	0	0	0	0	0	3
1	0	1	1	0	0	0	0	4
1	1	0	1	1	0	0	0	5
1	1	1	0	1	1	0	0	6
1	1	1	1	0	1	1	0	7
0	1	1	1	1	0	1	1	8
1	0	1	1	1	1	0	1	9
0	1	0	1	1	1	1	0	10
1	0	1	0	1	1	1	1	11
1	1	0	1	0	1	1	1	12
0	1	1	0	1	0	1	1	13
0	0	1	1	0	1	0	1	14
0	0	0	1	1	0	1	0	15
0	0	0	0	1	1	0	1	16
0	0	0	0	0	1	1	0	17
1	0	0	0	0	0	1	1	18
0	1	0	0	0	0	0	1	19
1	0	1	0	0	0	0	0	20
0	1	0	1	0	0	0	0	21
1	0	1	0	1	0	0	0	22
0	1	0	1	0	1	0	0	23
1	0	1	0	1	0	1	0	24
0	1	0	1	0	1	0	1	25
0	0	1	0	1	0	1	0	26
0	0	0	1	0	1	0	1	27
1	0	0	0	1	0	1	0	28
1	1	0	0	0	1	0	1	29
1	1	1	0	0	0	1	0	30
1	1	1	1	0	0	0	1	31
1	1	1	1	1	0	0	0	32
0	1	1	1	1	1	0	0	33
0	0	1	1	1	1	1	0	34
1	0	0	1	1	1	1	1	35
1	1	0	0	1	1	1	1	36
1	1	1	0	0	1	1	1	37
0	1	1	1	0	0	1	1	38
1	0	1	1	1	0	0	1	39
0	1	0	1	1	1	0	0	40
1	0	1	0	1	1	1	0	41
0	1	0	1	0	1	1	1	42
0	0	1	0	1	0	1	1	43

1	0	0	1	0	1	0	1	44
1	1	0	0	1	0	1	0	45
0	1	1	0	0	1	0	1	46
0	0	1	1	0	0	1	0	47
1	0	0	1	1	0	0	1	48
1	1	0	0	1	1	0	0	49
0	1	1	0	0	1	1	0	50
1	0	1	1	0	0	1	1	51
0	1	0	1	1	0	0	1	52
0	0	1	0	1	1	0	0	53
0	0	0	1	0	1	1	0	54
0	0	0	0	1	0	1	1	55
0	0	0	0	0	1	0	1	56
0	0	0	0	0	0	1	0	57
1	0	0	0	0	0	0	1	58
0	1	0	0	0	0	0	0	59
0	0	1	0	0	0	0	0	60
0	0	0	1	0	0	0	0	61
0	0	0	0	1	0	0	0	62
1	0	0	0	0	1	0	0	63
1	1	0	0	0	0	1	0	64
0	1	1	0	0	0	0	1	65
0	0	1	1	0	0	0	0	66
1	0	0	1	1	0	0	0	67
0	1	0	0	1	1	0	0	68
0	0	1	0	0	1	1	0	69
0	0	0	1	0	0	1	1	70
1	0	0	0	1	0	0	1	71
0	1	0	0	0	1	0	0	72
0	0	1	0	0	0	1	0	73
0	0	0	1	0	0	0	1	74
1	0	0	0	1	0	0	0	75
1	1	0	0	0	1	0	0	76
0	1	1	0	0	0	1	0	77
1	0	1	1	0	0	0	1	78
0	1	0	1	1	0	0	0	79
1	0	1	0	1	1	0	0	80
0	1	0	1	0	1	1	0	81
1	0	1	0	1	0	1	1	82
1	1	0	1	0	1	0	1	83
0	1	1	0	1	0	1	0	84
1	0	1	1	0	1	0	1	85
0	1	0	1	1	0	1	0	86
1	0	1	0	1	1	0	1	87
1	1	0	1	0	1	1	0	88
1	1	1	0	1	0	1	1	89
0	1	1	1	0	1	0	1	90
1	0	1	1	1	0	1	0	91
1	1	0	1	1	1	0	1	92
0	1	1	0	1	1	1	0	93
1	0	1	1	0	1	1	1	94

0	1	0	1	1	0	1	1	95
0	0	1	0	1	1	0	1	96
1	0	0	1	0	1	1	0	97
0	1	0	0	1	0	1	1	98
1	0	1	0	0	1	0	1	99
1	1	0	1	0	0	1	0	100
1	1	1	0	1	0	0	1	101
0	1	1	1	0	1	0	0	102
0	0	1	1	1	0	1	0	103
1	0	0	1	1	1	0	1	104
1	1	0	0	1	1	1	0	105
0	1	1	0	0	1	1	1	106
0	0	1	1	0	0	1	1	107
0	0	0	1	1	0	0	1	108
1	0	0	0	1	1	0	0	109
1	1	0	0	0	1	1	0	110
0	1	1	0	0	0	1	1	111
0	0	1	1	0	0	0	1	112
0	0	0	1	1	0	0	0	113
0	0	0	0	1	1	0	0	114
1	0	0	0	0	1	1	0	115
1	1	0	0	0	0	1	1	116
1	1	1	0	0	0	0	1	117
0	1	1	1	0	0	0	0	118
0	0	1	1	1	0	0	0	119
1	0	0	1	1	1	0	0	120
0	1	0	0	1	1	1	0	121
0	0	1	0	0	1	1	1	122
1	0	0	1	0	0	1	1	123
1	1	0	0	1	0	0	1	124
1	1	1	0	0	1	0	0	125
1	1	1	1	0	0	1	0	126
0	1	1	1	1	0	0	1	127
1	0	1	1	1	1	0	0	128
1	1	0	1	1	1	1	0	129
1	1	1	0	1	1	1	1	130
0	1	1	1	0	1	1	1	131
1	0	1	1	1	0	1	1	132
0	1	0	1	1	1	0	1	133
0	0	1	0	1	1	1	0	134
0	0	0	1	0	1	1	1	135
1	0	0	0	1	0	1	1	136
0	1	0	0	0	1	0	1	137
1	0	1	0	0	0	1	0	138
0	1	0	1	0	0	0	1	139
0	0	1	0	1	0	0	0	140
0	0	0	1	0	1	0	0	141
0	0	0	0	1	0	1	0	142
1	0	0	0	0	1	0	1	143
0	1	0	0	0	0	1	0	144
0	0	1	0	0	0	0	1	145

1	0	0	1	0	0	0	0	146
0	1	0	0	1	0	0	0	147
0	0	1	0	0	1	0	0	148
0	0	0	1	0	0	1	0	149
0	0	0	0	1	0	0	1	150
0	0	0	0	0	1	0	0	151
1	0	0	0	0	0	1	0	152
1	1	0	0	0	0	0	1	153
1	1	1	0	0	0	0	0	154
1	1	1	1	0	0	0	0	155
0	1	1	1	1	0	0	0	156
0	0	1	1	1	1	0	0	157
1	0	0	1	1	1	1	0	158
0	1	0	0	1	1	1	1	159
1	0	1	0	0	1	1	1	160
1	1	0	1	0	0	1	1	161
0	1	1	0	1	0	0	1	162
0	0	1	1	0	1	0	0	163
1	0	0	1	1	0	1	0	164
0	1	0	0	1	1	0	1	165
1	0	1	0	0	1	1	0	166
0	1	0	1	0	0	1	1	167
0	0	1	0	1	0	0	1	168
1	0	0	1	0	1	0	0	169
0	1	0	0	1	0	1	0	170
0	0	1	0	0	1	0	1	171
1	0	0	1	0	0	1	0	172
0	1	0	0	1	0	0	1	173
1	0	1	0	0	1	0	0	174
0	1	0	1	0	0	1	0	175
1	0	1	0	1	0	0	1	176
1	1	0	1	0	1	0	0	177
1	1	1	0	1	0	1	0	178
1	1	1	1	0	1	0	1	179
1	1	1	1	1	0	1	0	180
0	1	1	1	1	1	0	1	181
1	0	1	1	1	1	1	0	182
1	1	0	1	1	1	1	1	183
0	1	1	0	1	1	1	1	184
0	0	1	1	0	1	1	1	185
0	0	0	1	1	0	1	1	186
1	0	0	0	1	1	0	1	187
0	1	0	0	0	1	1	0	188
0	0	1	0	0	0	1	1	189
1	0	0	1	0	0	0	1	190
1	1	0	0	1	0	0	0	191
0	1	1	0	0	1	0	0	192
1	0	1	1	0	0	1	0	193
1	1	0	1	1	0	0	1	194
0	1	1	0	1	1	0	0	195
1	0	1	1	0	1	1	0	196

1	1	0	1	1	0	1	1	197
0	1	1	0	1	1	0	1	198
0	0	1	1	0	1	1	0	199
1	0	0	1	1	0	1	1	200
1	1	0	0	1	1	0	1	201
1	1	1	0	0	1	1	0	202
1	1	1	1	0	0	1	1	203
1	1	1	1	1	0	0	1	204
1	1	1	1	1	1	0	0	205
0	1	1	1	1	1	1	0	206
0	0	1	1	1	1	1	1	207
0	0	0	1	1	1	1	1	208
1	0	0	0	1	1	1	1	209
0	1	0	0	0	1	1	1	210
1	0	1	0	0	0	1	1	211
1	1	0	1	0	0	0	1	212
0	1	1	0	1	0	0	0	213
1	0	1	1	0	.	0	0	214
1	1	0	1	1	0	1	0	215
1	1	1	0	1	1	0	1	216
0	1	1	1	0	1	1	0	217
0	0	1	1	1	0	1	1	218
0	0	0	1	1	1	0	1	219
1	0	0	0	1	1	1	0	220
1	1	0	0	0	1	1	1	221
1	1	1	0	0	0	1	1	222
0	1	1	1	0	0	0	1	223
1	0	1	1	1	0	0	0	224
1	1	0	1	1	1	0	0	225
1	1	1	0	1	1	1	0	226
1	1	1	1	0	1	1	1	227
1	1	1	1	1	0	1	1	228
	1	1	1	1	1	0	1	229
1	1	1	1	1	1	1	0	230
0	1	1	1	1	1	1	1	231
1	0	1	1	1	1	1	1	232
0	1	0	1	1	1	1	1	233
0	0	1	0	1	1	1	1	234
1	0	0	1	0	1	1	1	235
1	1	0	0	1	0	1	1	236
1	1	1	0	0	1	0	1	237
0	1	1	1	0	0	1	0	238
0	0	1	1	1	0	0	1	239
0	0	0	1	1	1	0	0	240
0	0	0	0	1	1	1	0	241
1	0	0	0	0	1	1	1	242
0	1	0	0	0	0	1	1	243
1	0	1	0	0	0	0	1	244
1	1	0	1	0	0	0	0	245
1	1	1	0	1	0	0	0	246
1	1	1	1	0	1	0	0	247

```
0     1     1     1     1     0     1     0   248
0     0     1     1     1     1     0     1   249
0     0     0     1     1     1     1     0   250
0     0     0     0     1     1     1     1   251
0     0     0     0     0     1     1     1   252
0     0     0     0     0     0     1     1   253
0     0     0     0     0     0     0     1   254
0     0     0     0     0     0     0     0   255

test = 1   :   TOTAL = 32385   : check = 32385
```

APPENDIX B

A PAL22V10 was used as the BSG in both the transmitter and receiver of the experimental system. Originally, the firmware in the PAL22V10 used as the BSG in the receiver was modified to incorporate the integrate and dump pulse timing. This modification was successfully completed and tested, which verified that the required timing pulses could be generated without any additional hardware for a practical system which employs an appropriate programmable logic device as a BSG. For the experimental system, however, additional reference pulses were desired to provide bit comparison pulses at several locations within a received data bit. Since the PAL22V10 lacked the additional outputs required for the pulses, a EP-310 programmable logic device was connected with its inputs driven by the outputs of the original BSG. Since the additional bit comparison pulses would not be required in a practical system, the listing for an ABEL input program which may be used to program a PAL22V10 to generate both a length 255 m-sequence and the required timing pulses follows.

The resulting output equations and the resulting pin diagram for the device are also included.

ABEL INPUT FILE LISTING

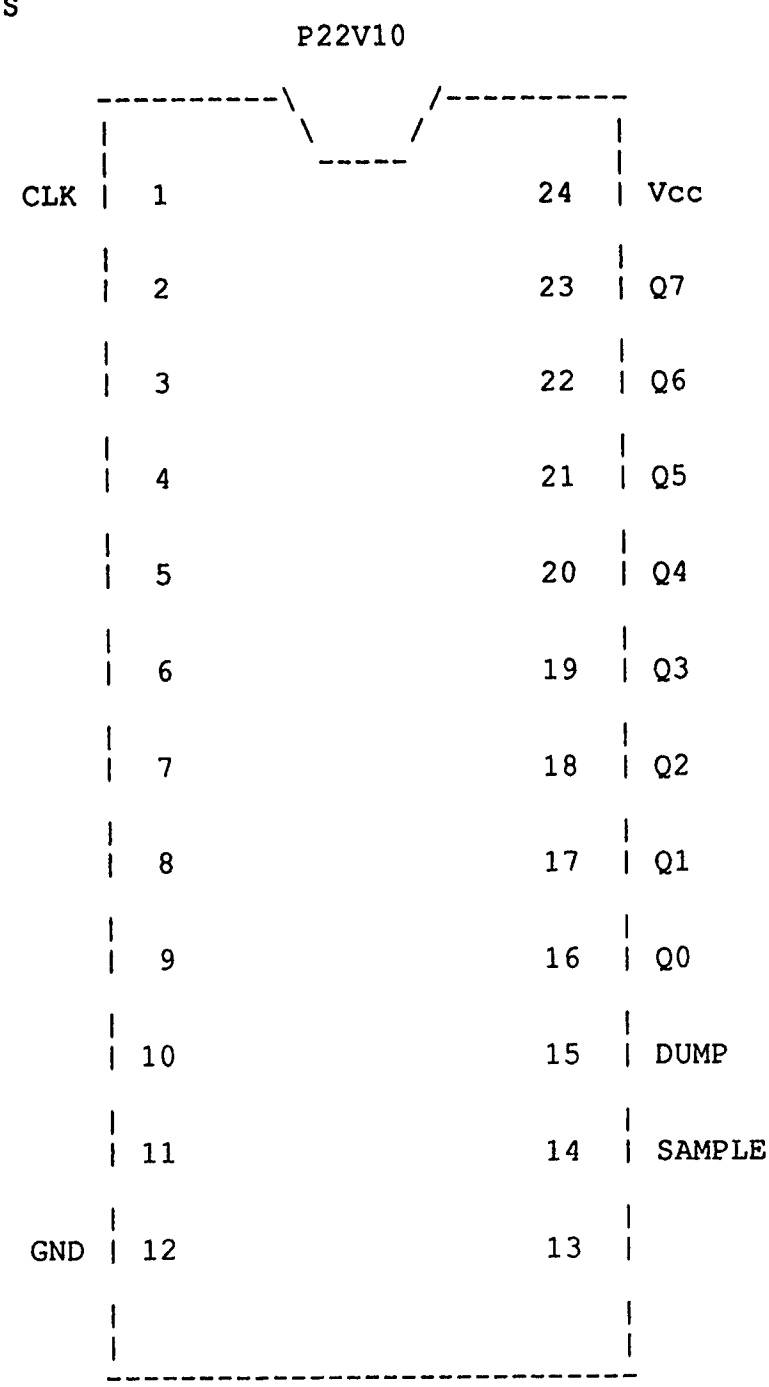
```
MODULE BSG

TITLE 'BSG CIRCUIT FROM HARSH DESIGN 10/20/88 WITH Add-ons'

    THESIS DEVICE 'p22v10';

"OUTPUT PINS

    Q7  PIN  23;
    Q6  PIN  22;
    Q5  PIN  21;
    Q4  PIN  20;
    Q3  PIN  19;
    Q2  PIN  18;
    Q1  PIN  17;
    Q0  PIN  16;
    DUMP   PIN  15;
    SAMPLE PIN  14;
    Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7 IsTYPE 'reg';
    DUMP,SAMPLE   IsTYPE 'com';


"INPUT PINS

    CLK    PIN  1;

EQUATIONS

" NOTE: The output pin numbers differ from those generated by
"MATLAB program in Appendix A, because they are referenced 0
"through 7 vice 1 through 8.

Q0 := ((Q7 !$ Q3) !$ (Q1 !$ Q2));
Q1 := Q0;
Q2 := Q1;
Q3 := Q2;
Q4 := Q3;
Q5 := Q4;
Q6 := Q5;
Q7 := Q6;
DUMP = !Q0 & !Q1 & !Q2 & Q3 & !Q4 & Q5 & Q6 & Q7 #
       !Q0 & !Q1 & Q2 & !Q3 &  Q4 &  Q5 & Q6 & !Q7 ;
SAMPLE = Q0 & Q1 & !Q2 & Q3 & Q4 & Q5 & Q6 & Q7 ;

END BSG
```

```
ABEL(tm) Version 2.00b  -  Document Generator
29-Apr-90 09:32 AM

Equations for Module BSG

Device THESIS


     Reduced Equations:

       Q0 := (!Q1 & !Q2 & !Q3 & !Q7
             # !Q1 & !Q2 & Q3 & Q7
             # !Q1 & Q2 & !Q3 & Q7
             # !Q1 & Q2 & Q3 & !Q7
             # Q1 & !Q2 & !Q3 & Q7
             # Q1 & !Q2 & Q3 & !Q7
             # Q1 & Q2 & !Q3 & !Q7
             # Q1 & Q2 & Q3 & Q7);


       Q1 := (Q0);

       Q2 := (Q1);

       Q3 := (Q2);

       Q4 := (Q3);

       Q5 := (Q4);

       Q6 := (Q5);

       Q7 := (Q6);

       DUMP = (!Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7
              # !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & Q7);


       SAMPLE = (!Q0 & !Q1 & !Q2 & !Q3 & Q4 & Q5 & Q6 & !Q7);
```

ABEL(tm) Version 2.00b - Document Generator
29-Apr-90 09:32 AM

Chip diagram for Module BSG

Device THESIS



End of module BSG

The ABLE input file for the EP-310 used as part of the experimental system follows. This file may be used to generate the timing information when the parallel outputs of the BSG are available from another PAL device or a set of discrete logic gates as used in many direct-sequence spread spectrum receivers.

```
MODULE FIX310

    TEMPFIX DEVICE 'E0310';

"OUTPUT PINS

    DUMP       PIN  15;
    SAMPLE     PIN  14;
    BITCOMP1   PIN  12;
    BITCOMP2   PIN  13;
    BITCOMP3   PIN  18;
    BITCOMP4   PIN  19;

"INPUT PINS

    Q7  PIN  2;
    Q6  PIN  3;
    Q5  PIN  4;
    Q4  PIN  5;
    Q3  PIN  6;
    Q2  PIN  7;
    Q1  PIN  8;
    Q0  PIN  9;

EQUATIONS

DUMP = !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7 #
       !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & Q7 #
       Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7 #
       Q0 & Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7 #
       !Q0 & Q1 & Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7;
"       Q0 & !Q1 & Q2 & Q3 & !Q4 & !Q5 & !Q6 & !Q7 #
"       Q0 & Q1 & !Q2 & Q3 & Q4 & !Q5 & !Q6 & !Q7 ;
SAMPLE = !Q0 & !Q1 & !Q2 & !Q3 & Q4 & Q5 & Q6 & Q7 ;
BITCOMP1 = !Q0 & Q1 & !Q2 & Q3 &  Q4 &  Q5 & Q6 & !Q7 ;
BITCOMP2 = Q0 & Q1 & !Q2 & Q3 & !Q4 & !Q5 & Q6 & !Q7 ;
BITCOMP3 = Q0 & !Q1 & !Q2 & Q3 &  Q4 & !Q5 & Q6 & Q7 ;
BITCOMP4 = !Q0 & !Q1 & !Q2 & Q3 & Q4 & Q5 & !Q6 & !Q7 ;
```

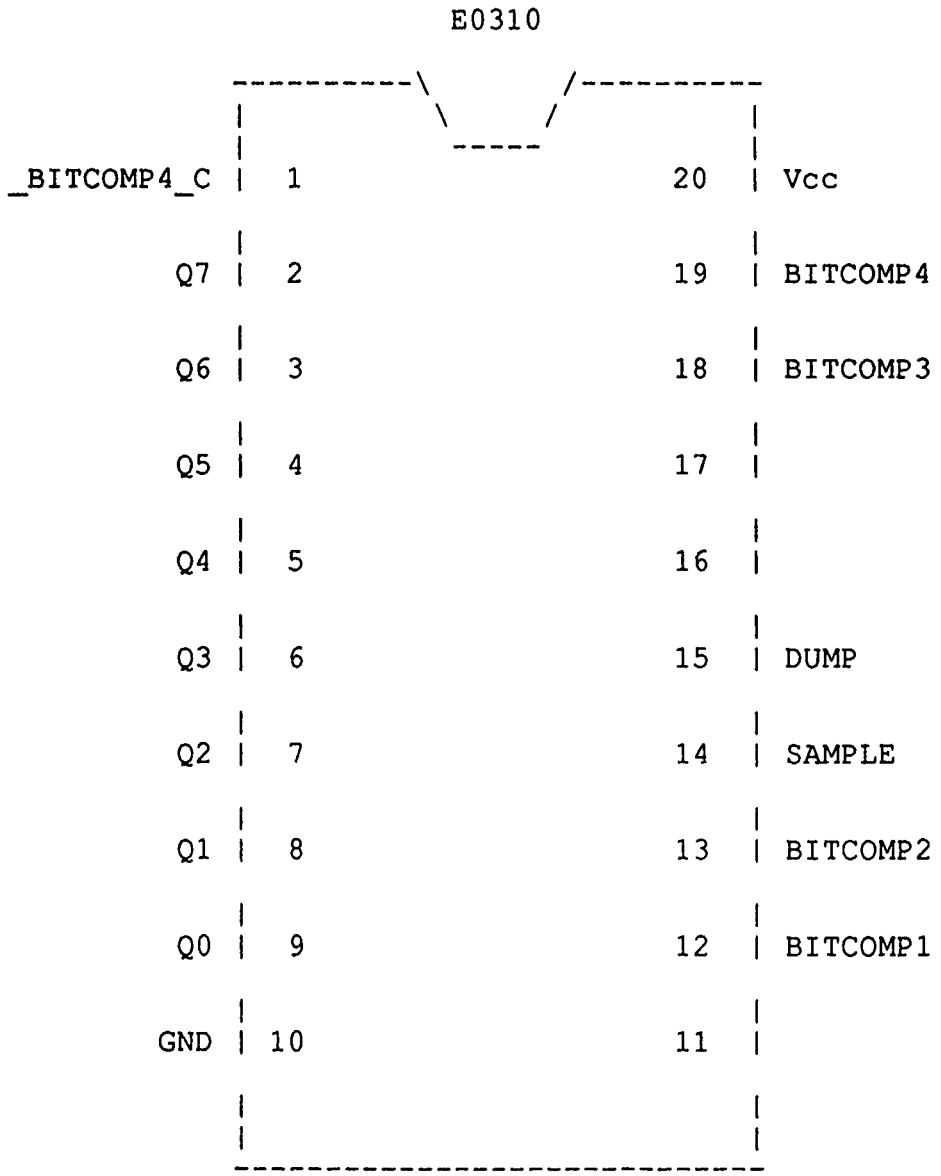
The resulting output equations and pin diagram for the EP-310 follow.

```
ABEL(tm) Version 2.00b  -  Document Generator
08-May-90 02:42 PM

Equations for Module FIX310

Device TEMPFIX

     Reduced Equations:

       DUMP = (!Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7
              # !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & Q7
              # !Q0 & Q1 & Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7
              # Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7
              # Q0 & Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7);


       SAMPLE = (!Q0 & !Q1 & !Q2 & !Q3 & Q4 & Q5 & Q6 & Q7);

       BITCOMP1 = (!Q0 & Q1 & !Q2 & Q3 & Q4 & Q5 & Q6 & !Q7);

       BITCOMP2 = (Q0 & Q1 & !Q2 & Q3 & !Q4 & !Q5 & Q6 & !Q7);

       BITCOMP3 = (Q0 & !Q1 & !Q2 & Q3 & Q4 & !Q5 & Q6 & Q7);

       BITCOMP4 = (!Q0 & !Q1 & !Q2 & Q3 & Q4 & Q5 & !Q6 &
!Q7);
```

ABEL(tm) Version 2.00b - Document Generator
08-May-90 02:42 PM

Chip diagram for Module FIX310

Device TEMPFIX



end of module FIX310

APPENDIX C

The improvement in signal-to-noise ratio due to despreading in the receiver can be described in terms of Processing Gain. By definition, the Processing Gain (PG) of a system is

$$PG = \frac{OutputSNR}{InputSNR} \qquad (1)$$

Direct-sequence spread spectrum systems are used because they provide PG that is a direct function of the ratio of radio frequency (RF) bandwidth to the data bit rate. Since the RF bandwidth is related to the data bit rate in terms of the length of the spreading sequence, it may be concluded that [Ref. 2: p.25]

$$PG = \frac{(0.88)\ ClockRate}{DataRate} = (0.88)L \qquad (2)$$

where L is the length of the spreading sequence.
Thus for the experimental system,

$$PG = 10\log[(0.88)L] = 10\log[(0.88)255] = 23.5dB.$$

Processing Gain provides an improvement in system performance against many interfering signals. However, due to the increase in bandwidth at the transmitter, there is no net

increase in system performance when the receiver is limited by thermal noise.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Commander Naval Air Test Center RD-RTG Patuxent River, MD 20670	2
3.	Library Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
4.	Department Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Professor Glen Myers, Code EC/Mv Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	5
6.	Professor Tri Ha, Code EC/Ha Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
7.	Stuart D. Harshbarger, Electronics Engineer Naval Air Test Center Research and Technology Group, Range Directorate Patuxent River, MD 20670	2